



APPENDIX A

```

/*****
 *
 Copyright Notice: This source code is (C) Copyright 1996, EarthWeb LLC.
 ,
 All Rights Reserved. Distribution of this document or it's resulting
 compiled code is granted for non-commercial use, with prior approval of
 EarthWeb LLC. Distribution of this document or its resulting compiled
 code, for commercial use, is granted only with prior written approval o
 f
 EarthWeb, LLC. For information, send email to info@earthweb.com.
 *****/
/
public class nativeInputStream extends java.io.InputStream {
    public native int read();
    public native void nativeinit();
    public nativeInputStream() {
        super();
        nativeinit();
    }
    static {
        System.loadLibrary("instream");
    }
}

```

```

/*****
*
Copyright Notice: This source code is (C) Copyright 1996, EarthWeb LLC.
All Rights Reserved. Distribution of this document or it's resulting
compiled code is granted for non-commercial use, with prior approval of
EarthWeb LLC. Distribution of this document or its resulting compiled
code, for commercial use, is granted only with prior written approval o
f
EarthWeb, LLC. For information, send email to info@earthweb.com.
*****/
/
#include <StubPreamble.h>
#include "nativeInputStream.h"
#include <Windef.h>
#include <Winbase.h>
#include <stdio.h>

HANDLE h;

long nativeInputStream_read(struct HnativeInputStream *) {
    DWORD evtype;
    unsigned long nread = 0; // the number of characters read
    unsigned char result; // the character read
    while (nread == 0) ReadFile(h, (LPVOID)&result, 1, &nread, NULL);
    return (long)result;
}

void nativeInputStream_nativeinit(struct HnativeInputStream *) {
    h = CreateFile("COM1", GENERIC_READ|GENERIC_WRITE, 0, NULL,
        OPEN_EXISTING, 0, NULL);

    DCB dcb;
    GetCommState(h, &dcb);
    dcb.BaudRate = 9600;
    dcb.ByteSize = 8;
    dcb.Parity = NOPARITY;
    dcb.StopBits = ONESTOPBIT;
    SetCommState(h, &dcb);
    SetCommMask(h, EV_RXCHAR);
    char *comcode = "\1B";
    unsigned long nwritten = 0;
    while (nwritten < 2) WriteFile(h, (LPVOID)comcode, 2, &nwritten, NULL
    );
    printf("Yeah!");
}

```

```

/*****
*
Copyright Notice: This source code is (C) Copyright 1996, EarthWeb LLC.
All Rights Reserved. Distribution of this document or it's resulting
compiled code is granted for non-commercial use, with prior approval of
EarthWeb LLC. Distribution of this document or its resulting compiled
code, for commercial use, is granted only with prior written approval o
f
EarthWeb, LLC. For information, send email to info@earthweb.com.
*****/
/
#include <Wintypes.h>
#include <Winbase.h>
#include <Windef.h>
#include <stdio.h>
#include <conio.h>

// #define COM1 = 0x3F8
// #define COM2 = 0x2F8

HANDLE h;
int main() {
    unsigned char result;
    unsigned long nread;

    h = CreateFile("COM1", GENERIC_READ|GENERIC_WRITE, 0, NULL,
        OPEN_EXISTING, 0, NULL);
    if (h == INVALID_HANDLE_VALUE) {
        printf("NOOOOO!");
    }
    DCB dcb;
    GetCommState(h, &dcb);
    dcb.BaudRate = 9600;
    dcb.ByteSize = 8;
    dcb.Parity = NOPARITY;
    dcb.StopBits = ONESTOPBIT;
    SetCommState(h, &dcb);

    SetCommMask(h, EV_RXCHAR);
    printf("Reading serial port:\n");

    DWORD evtttype;
    while (1) {
        printf("%c", nativeInputSteam_read() {
            WaitCommEvent(h, &evtttype, NULL);
            while (ReadFile(h, (LPVOID)&result, 1, &nread, NULL))
                printf("%c", result); */
        }
    }
    return 0;
}

```

```
long nativeInputStream_read() {  
    DWORD evttype;  
    unsigned long nread; // the number of characters read  
    unsigned char result; // the character read  
    WaitCommEvent(h, &evttype, NULL);  
    ReadFile(h, (LPVOID)&result, 1, &nread, NULL);  
    return (long)result;  
}
```

```

/*****
 *
 Copyright Notice: This source code is (C) Copyright 1996, EarthWeb LLC.
 All Rights Reserved. Distribution of this document or it's resulting
 compiled code is granted for non-commercial use, with prior approval of
 EarthWeb LLC. Distribution of this document or its resulting compiled
 code, for commercial use, is granted only with prior written approval o
 f
 EarthWeb, LLC. For information, send email to info@earthweb.com.
 *****/
/

```

```

import java.applet.*;
import java.awt.*;
import java.io.*;
import java.net.URL;
import java.net.MalformedURLException;
import java.util.Hashtable;

class blackPanel extends Panel {
    public void paint(Graphics g) {
        g.setColor(Color.black);
        g.fillRect(0, 0, size().width, size().height);
    }
}

class displayFrame extends Panel implements Runnable {
    int xoffset = 30;
    int yoffset = 40;

    List URLlist;
    // TextField text1;
    Checkbox autobrowse;
    String text1Str = "";
    int rlx, rly;
    AppletContext apc;

    Thread thread;
    InputStream din;
    Hashtable dict;
    Hashtable visited;

    boolean periodbefore; // HACK ALERT

    public displayFrame(String frameName, AppletContext apc, InputS
tream in) {
        //      super(frameName);
        this.apc = apc;
        din = in;
        dict = new Hashtable();
        visited = new Hashtable();

```

```

thread = new Thread(this);
thread.start();

        resize(106 + xoffset, 156 + yoffset);

        rlx = -(size().width/4);
        rly = size().height/4;

        setLayout( new BorderLayout() );
        add( "South", autobrowse = new Checkbox("Autobrowse"));
        autobrowse.setState(true);

        URLlist = new List();
        add("Center", URLlist);
        //rl.setConstraint(URLlist, autobrowse, RelativeLayout.
under, -4, 10);
        //rl.setConstraint(URLlist, button0, RelativeLayout.und
er, -4, 10);
        //
        URLlist.setBackground( Color.cyan );
        Font newfont = new Font("TimesRoman", Font.PLAIN, 10);
        //
        URLlist.setFont( newfont );

    }

    public synchronized void turnToURL(String s, String target) {
        System.out.println("Turning to URL " + s + target);
        visited.put(s, this);
        try apc.showDocument(new java.net.URL(s), target);
        catch (Exception e);
    }

    public void turnToTitle(String s, String target) {
        System.out.println("Turning to Title " + s + target);
        turnToURL((String)(dict.get(s)), target);
    }

    public synchronized void addURL(String title, String url) {
        String previous = (String)(dict.get(title));
        if (!(dict.containsKey(title))) URLlist.addItem(title);
        dict.put(title, url);
    }

    public void run() {

        Thread.currentThread().setPriority( Thread.MIN_PRIORITY );
        //System.out.println("Context: "+apcon);
        char buf[] = new char[128];
        int bufidx = 0;
        boolean capture = false;
        String currentTitle = null;

        try {

```

```

// Listener loop

int c;
periodbefore = false;
while ((c = (din.read())) != -1) {
    if (c == '.') {
        if (capture && periodbefore) {
            bufidx--;
            c = (byte) '>';
            periodbefore = false;
        } else periodbefore = true;
    } else periodbefore = false;
    if ((char)c == '>') {
        capture = false;
        String result = new String(buf, 0, bufi
dx);
        System.out.println("Remote found tag "+
result);
        if (result.startsWith("URL:")) {
            result = result.substring(4);
            if (currentTitle != null) addURL(curr
entTitle, result);
            if (autobrowse.getState() && !(visite
d.containsKey(result))) {
                URLlist.select(URLlist.countIte
ms()-1);
                turnToURL(result, "browser");
            }
            currentTitle = null;
        }
        else if (result.startsWith("Title:"))
            currentTitle = result.substring(6);
    }
    if (capture) {
        buf[bufidx++] = (char)c;
        if (bufidx >= buf.length)
            capture = false;
    }
    if ((char)c == '<') {
        capture = true;
        bufidx = 0;
    }
}
// This is an acceptable exception
} catch( EOFException e ) System.out.println( "URLClient: eof" );
// This is probably not.
catch( IOException e ) System.out.println( "URLClient.run: "+e );
}

// public void paint(Graphics g) {

```


TVcontrol.java

```
//          g.setColor(Color.black);
//          g.fillRect(0, 0, size().width, size().height);
//      }

      public void update(Graphics g) {
          repaint();
      }

      public boolean action(Event evt, Object what) {
          URL where;

          turnToTitle((String)what, "browser");
          repaint();
          return true;
      }
  }

  public class TVcontrol extends Applet {
      static displayFrame remoteFrame = null;

      public void init() {
          InputStream in;
          AppletContext apc = getAppletContext();

          in = new nativeInputStream();
          setLayout(new BorderLayout());

          if (remoteFrame == null)
              remoteFrame = new displayFrame("Remote Control"
, apc, in);

          add("Center", remoteFrame);
          remoteFrame.show();
      }
      public void start() {
          remoteFrame.apc = getAppletContext();
      }
  }
```

```
<TITLE>WebTV</TITLE>
```

```
<FRAMESET ROWS="33,66">
```

```
  <FRAMESET COLS="35,65">
```

```
    <FRAME SCROLLING="no" NAME="tv" SRC="file://once-ler/pub/moo.tv
```

```
">
```

```
    <FRAME SCROLLING="no" NAME="control" SRC="control.html">
```

```
      <NOFRAMES>blank.html</NOFRAMES>
```

```
  </FRAMESET>
```

```
  <FRAME SCROLLING="yes" NAME="browser" SRC="browser.html">
```

```
</FRAMESET>
```

```
<title>Remote</title>
```

```
<center>
```

```
<applet code="TVcontrol.class" width=300 height=100></applet>
```

```
</center>
```

```
<title></title>
```

```
<body bgcolor="#ffffff">
```

```
<center><b>This is the browser window.</b></center>
```

```

/*****
*
Copyright Notice: This source code is (C) Copyright 1996, EarthWeb LLC.
All Rights Reserved. Distribution of this document or it's resulting
compiled code is granted for non-commercial use, with prior approval of
EarthWeb LLC. Distribution of this document or its resulting compiled
code, for commercial use, is granted only with prior written approval o
f
EarthWeb, LLC. For information, send email to info@earthweb.com.
*****/
/
#ifndef __INST_H__
#define __INST_H__

#include "npapi.h"

class CPluginWindow;

//
// Instance state information about the plugin.
//
// *Developers*: Use this struct to hold per-instance
//                information that you'll need in the
//                various functions in this file.
//
typedef struct _PluginInstance
{
    NPWindow*      fWindow;
    uint16         fMode;
    CPluginWindow* window;
} PluginInstance;

#endif

```

```

/*****
*
Copyright Notice: This source code is (C) Copyright 1996, EarthWeb LLC.
All Rights Reserved. Distribution of this document or it's resulting
compiled code is granted for non-commercial use, with prior approval of
EarthWeb LLC. Distribution of this document or its resulting compiled
code, for commercial use, is granted only with prior written approval o
f
EarthWeb, LLC. For information, send email to info@earthweb.com.
*****/
/
#include "windows.h"
#ifndef _NPAPI_H_
#include "npapi.h"
#endif
#include "inst.h"
#include "plgwnd.h"
#include "wintv.h"

//-----
// NPP_Initialize:
//-----
NPPError NPP_Initialize(void)
{
    PCV_Initialize();
    return NPERR_NO_ERROR;
}

//-----
// NPP_Shutdown:
//-----
void NPP_Shutdown(void)
{
    WTV_Exit(0);
}

//-----
// NPP_New:
//-----
NPPError NP_LOADDS
NPP_New(NPMIMEType pluginType,

```

```

        NPP instance,
        uint16 mode,
        int16 argc,
        char* argn[],
        char* argv[],
        NP SavedData* saved)
{
    if (instance == NULL)
        return NPERR_INVALID_INSTANCE_ERROR;

    instance->pdata = NPN_MemAlloc(sizeof(PluginInstance));
    PluginInstance* This = (PluginInstance*) instance->pdata;
    if (This != NULL)
    {
        This->fWindow = NULL;
        This->window = NULL;
        This->fMode = mode;    // Mode is NP_EMBED, NP_FULL, or NP_BACK
GROUND (see npapi.h)

        instance->pdata = This; // save my data pointer in the
instance pdata pointer          // this will be passed back to me in a
ll calls so that I              // can extract it later

        return NPERR_NO_ERROR;
    }
    else
        return NPERR_OUT_OF_MEMORY_ERROR;
}

static void UnSubclass(PluginInstance *This)
{
    WNDPROC      OldWndProc;
    WNDPROC*      lplpfn = This->window->GetSuperWndProcAddr();

    if ( !*lplpfn )
    {
        ASSERT(0);
        return;
    }

    // Set the original window procedure
    OldWndProc = (WNDPROC)::SetWindowLong( This->window->m_hWnd,

        GWL_WNDPROC, (LONG) *lplpfn );

    // A subclassed window's procedure is always AfxWndProc.
    // If this is not TRUE, then it's not a subclassed window.

```

```

    if ( OldWndProc != AfxWndProc )
        ASSERT(0);
}

static void KillTvWindow(PluginInstance *This)
{
    /* if (This->cAvi) {
        This->cAvi->Close();
        delete This->cAvi;
        This->cAvi = NULL;
    } */
    UnSubclass(This);
    if (This->window) {
        This->window->Detach();
        delete This->window;
        This->window = NULL;
    }
}

//-----
// NPP_Destroy:
//-----
NPError NP_LOADDS
NPP_Destroy(NPP instance, NPSavedData** save)
{
    if (instance == NULL)
        return NPERR_INVALID_INSTANCE_ERROR;

    PluginInstance* This = (PluginInstance*) instance->pdata;

    if (This != NULL)
    {
        KillTvWindow(This);
        NPN_MemFree(instance->pdata);
        instance->pdata = NULL;
    }

    return NPERR_NO_ERROR;
}

//-----
// NPP_SetWindow:
//-----

```



```

NPErrror NP_LOADDDS
NPP_SetWindow(NPP instance, NPWindow* window)
{
    if (instance == NULL)
        return NPERR_INVALID_INSTANCE_ERROR;

    PluginInstance* This = (PluginInstance*) instance->pdata;

    if (!window)
        return NPERR_GENERIC_ERROR;

    if (!instance)
        return NPERR_INVALID_INSTANCE_ERROR;

    if (!This)
        return NPERR_GENERIC_ERROR;

    if (!window->window && !This->window) // spurious entry
        return NPERR_NO_ERROR;

    if (!window->window && This->window)
    {
        // window went away
        KillTvWindow(This);
        return NPERR_NO_ERROR;
    }

    if (!This->window && window->window) {
        This->window = (CPluginWindow *) new CPluginWindow();
        if (!This->window->SubclassWindow((HWND)window->window)) {
            MessageBox(NULL, "SubclassWindow Failed", "Plug-in-test", MB_O
K);
            return NPERR_GENERIC_ERROR;
        }

        PCV_EnableColorKey();
        WTV_ResetAudio();
        This->window->StoreData(This);
    }

    This->window->InvalidateRect(NULL);
    This->window->UpdateWindow();

    RECT r;
    GetWindowRect((HWND)window->window, &r);
    PCV_DisableVideo();
    PCV_CreateWindow(r.left, r.top, window->width, window->height,
1);
    PCV_EnableVideo();
    WTV_CheckQFactor();

    // ((HWND)window->window)->hbrBackground = This->bkbrush;
    // InvalidateRect((HWND)window->window, NULL, TRUE);

    This->fWindow = window;

```

```

    return NPERR_NO_ERROR;
}

//-----
// NPP_NewStream:
//-----
NPError NP_LOADDS
NPP_NewStream(NPP instance,
               NPMIMEType type,
               NPStream *stream,
               NPBool seekable,
               uint16 *styp)
{
    if (instance == NULL)
        return NPERR_INVALID_INSTANCE_ERROR;
    PluginInstance* This = (PluginInstance*) instance->pdata;
    return NPERR_NO_ERROR;
}

int32 STREAMBUFSIZE = 0X0FFFFFFF; // If we are reading from a file in
NPAsFile                          // mode so we can take any size str
eam in our                        // write call (since we ignore it)

//-----
// NPP_WriteReady:
//-----
int32 NP_LOADDS
NPP_WriteReady(NPP instance, NPStream *stream)
{
    if (instance != NULL)
    {
        PluginInstance* This = (PluginInstance*) instance->pdata;

        return STREAMBUFSIZE; // Number of bytes ready to accept in NPP_W
rite()
    }
}

```

```

//-----
// NPP_Write:
//-----
int32 NP_LOADDS
NPP_Write(NPP instance, NPStream *stream, int32 offset, int32 len, void
*buffer)
{
    if (instance != NULL)
    {
        PluginInstance* This = (PluginInstance*) instance->pdata;

        return len;                // The number of bytes accepted
    }
}

//-----
// NPP_DestroyStream:
//-----
NPError NP_LOADDS
NPP_DestroyStream(NPP instance, NPStream *stream, NPError reason)
{
    if (instance == NULL)
        return NPERR_INVALID_INSTANCE_ERROR;
    PluginInstance* This = (PluginInstance*) instance->pdata;

    return NPERR_NO_ERROR;
}

//-----
// NPP_StreamAsFile:
//-----
void NP_LOADDS
NPP_StreamAsFile(NPP instance, NPStream *stream, const char* fname)
{
    if (instance != NULL)
    {
        PluginInstance* This = (PluginInstance*) instance->pdata;
    }
}

```

```

}

//-----
// NPP_Print:
//-----
void NP_LOADDS
NPP_Print(NPP instance, NPPrint* printInfo)
{
    if(printInfo == NULL)    // trap invalid parm
        return;

    if (instance != NULL)
    {
        PluginInstance* This = (PluginInstance*) instance->pdata;

        if (printInfo->mode == NP_FULL)
        {
            void* platformPrint = printInfo->print.fullPrint.platformPr
int;
            NPBool printOne = printInfo->print.fullPrint.printOne;

            printInfo->print.fullPrint.pluginPrinted = FALSE; // Do the
default

        }
        else    // If not fullscreen, we must be embedded
        {
            NPWindow* printWindow = &(printInfo->print.embedPrint.windo
w);
            void* platformPrint = printInfo->print.embedPrint.platformP
rint;
        }
    }
}

//-----
// NPP_HandleEvent:
// Mac-only.
//-----
int16 NPP_HandleEvent(NPP instance, void* event)
{
    NPBool eventHandled = FALSE;
    if (instance == NULL)

```

```
        return eventHandled;

PluginInstance* This = (PluginInstance*) instance->pdata;

return eventHandled;
}
```

```

/*****
*
Copyright Notice: This source code is (C) Copyright 1996, EarthWeb LLC.
All Rights Reserved. Distribution of this document or it's resulting
compiled code is granted for non-commercial use, with prior approval of
EarthWeb LLC. Distribution of this document or its resulting compiled
code, for commercial use, is granted only with prior written approval o
f
EarthWeb, LLC. For information, send email to info@earthweb.com.
*****/
/
#include "stdafx.h"
#include "plgwnd.h"
#include <mmsystem.h>
#include "wintv.h"

// CPluginWindow constructor:
//
CPluginWindow::CPluginWindow()
{
    m_data = NULL;
}

void CPluginWindow::OnPaint() {
    CPaintDC dc(this); // device context for painting

    //CWnd::OnPaint();
    CBrush brush(WTV_GetKeyColor());
    dc.FillRect(&(dc.m_ps.rcPaint), &brush);
}

/*void CPluginWindow::OnMove(int x, int y) {
    POINT worg;
    PCV_DisableVideo();
    PCV_SetWindowPosition((WORD)worg.x, (WORD)worg.y);
    WTV_CheckQFactor();
    PCV_EnableVideo();
} */

void CPluginWindow::StoreData(PluginInstance *data) {
    m_data = data;
}

//////////
// Return place to hold original window proc
WNDPROC* CPluginWindow::GetSuperWndProcAddr()
{
#ifdef WIN32
    return CWnd::GetSuperWndProcAddr();
#else
    static WNDPROC pfnSuper; // place to store window proc
#endif
}

```

```
        return &pfnSuper;           // always return the same address
    #endif
}

// CMainWindow message map:
// Associate messages with member functions.
//
// It is implied that the ON_WM_PAINT macro expects a member function
// "void OnPaint()".
//
BEGIN_MESSAGE_MAP( CPluginWindow, CWnd )
    //{AFX_MSG_MAP( CMainWindow )
    ON_WM_PAINT()
    ON_WM_MOVE()
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()
```

```

/*****
*
Copyright Notice: This source code is (C) Copyright 1996, EarthWeb LLC.
All Rights Reserved. Distribution of this document or it's resulting
compiled code is granted for non-commercial use, with prior approval of
EarthWeb LLC. Distribution of this document or its resulting compiled
code, for commercial use, is granted only with prior written approval o
f
EarthWeb, LLC. For information, send email to info@earthweb.com.
*****/
/
#ifndef __PLGWND_H__
#define __PLGWND_H__

#include "npapi.h"
#include "stdafx.h"

#include "inst.h"

////////////////////////////////////
////////

// CPluginWindow:
//
class CPluginWindow : public CWnd
{
protected:
    PluginInstance * m_data;
    // CWnd * m_videoWnd;
public:
    CPluginWindow();
    void StoreData(PluginInstance *);
    virtual WNDPROC* GetSuperWndProcAddr();

    //{AFX_MSG( CMainWindow )
    afx_msg void OnPaint();
    //
    afx_msg void OnMove(int x, int y);
    //}}AFX_MSG

    DECLARE_MESSAGE_MAP()
};

#endif

```



```
// PLUGTEST.RC2 - resources Microsoft Visual C++ does not edit directly
//
```

```
#ifndef APSTUDIO_INVOKED
    #error this file is not editable by Microsoft Visual C++
#endif //APSTUDIO_INVOKED
```

```
////////////////////////////////////
//////
//
// Version
//
```

```
VS_VERSION_INFO VERSIONINFO
```

```
FILEVERSION 1,0,0,1
```

```
PRODUCTVERSION 1,0,0,1
```

```
FILEFLAGSMASK 0x3fL
```

```
#ifdef _DEBUG
```

```
FILEFLAGS 0x1L
```

```
#else
```

```
FILEFLAGS 0x0L
```

```
#endif
```

```
FILEOS 0x4L
```

```
FILETYPE 0x2L
```

```
FILESUBTYPE 0x0L
```

```
BEGIN
```

```
    BLOCK "StringFileInfo"
```

```
    BEGIN
```

```
        BLOCK "040904e4"
```

```
        BEGIN
```

```
            VALUE "CompanyName",      "Earthweb LLC\0"
            VALUE "FileDescription",    "Win/TV Inline Plugin\0"
            VALUE "FileVersion",        "0.0.0.1\0"
            VALUE "InternalName",       "Inline TV\0"
            VALUE "LegalCopyright",     "Copyright Earthweb LLC 1996\0"
            VALUE "LegalTrademarks",    "\0"
            VALUE "OriginalFilename",   "npd1132.dll\0"
            VALUE "ProductName",        "Inline TV Plugin\0"
            VALUE "ProductVersion",     "0.0.0.1\0"
            VALUE "MIMEType",           "fake/mime-type\0"
            VALUE "FileExtents",        "tv\0"
            VALUE "FileOpenName",       "Fake MIME (*.tv)\0"
```

```
        END
```

```
    END
```

```
    BLOCK "VarFileInfo"
```

```
    BEGIN
```

```
        VALUE "Translation", 0x409, 1252
```

```
    END
```

```
END
```

////////////////////////////////////

//////

// Add manually edited resources here...

////////////////////////////////////

//////